

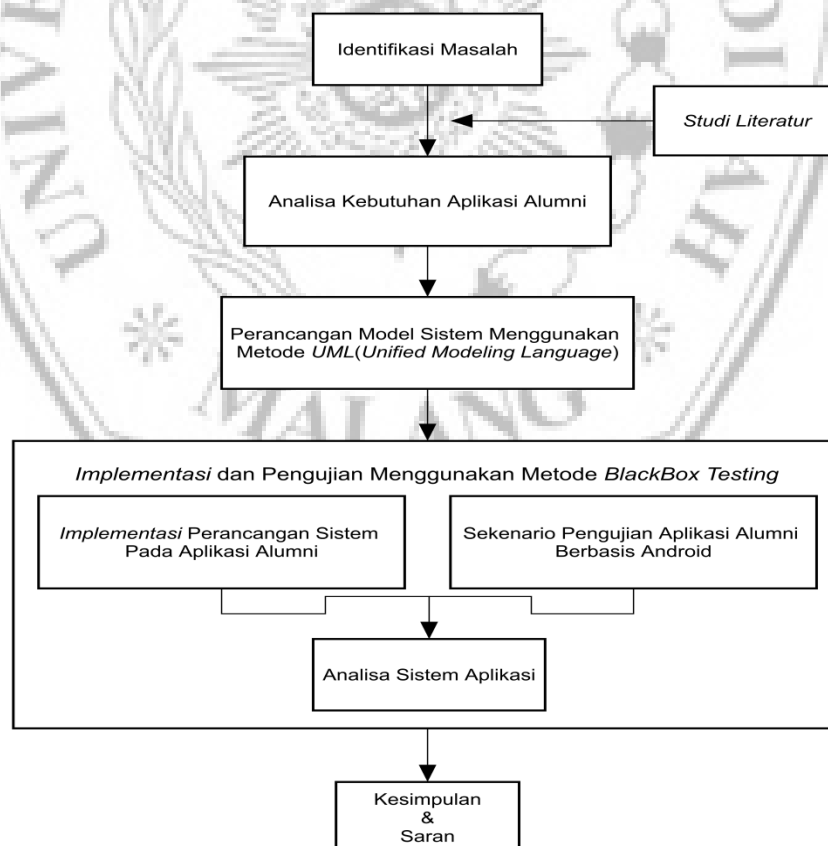
## BAB III

### PERANCANGAN SISTEM

Pada bab ini akan dibahas dan dijelaskan dengan spesifik mengenai perancangan sistem dalam pembuatan aplikasi alumni berbasis android. Aplikasi ini merupakan fasilitas yang memberikan kemudahan kepada alumni Jurusan Teknik Elektro Universitas Muhammadiyah Malang untuk bertukar pikiran menyampaikan informasi terbaru. Sistem yang dibangun merupakan sebuah aplikasi *mobile* yang dapat diakses kapanpun dan dimanapun.

#### 3.1 Metodologi

Metodologi yang digunakan dalam melaksanakan penelitian ini merupakan dasar penyusunan rancangan penelitian dan merupakan penjabaran dari metode ilmiah secara umum. Penyelesaian masalah dalam tugas akhir ini dengan menggunakan metodologi sebagai berikut :



Gambar 3.1 Metodologi Penelitian

Pada gambar 3.1 telah ditunjukkan metodologi yang digunakan meliputi beberapa tahapan, yaitu analisa, desain, *implementasi* dan *testing*. Setiap tahapan akan dijelaskan dalam sub bab sebagai berikut :

### 3.2 Studi Literatur

#### ✓ Studi Literatur

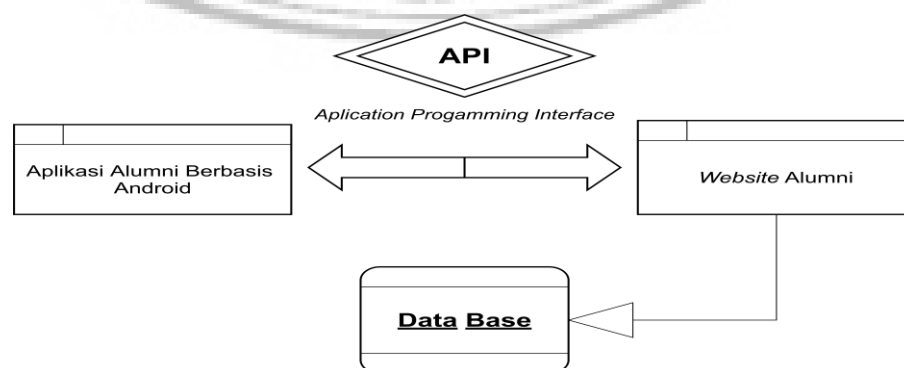
Pada tahap ini dilakukan pendalaman *e-book*, jurnal, internet, dan *literatur* lainnya yang berhubungan dengan rancang bangun aplikasi alumni.

### 3.3 Pengumpulan Data

Tahap berikutnya yaitu melakukan pengumpulan data yang akan digunakan untuk proses perancangan data aplikasi. Data yang dikumpulkan merupakan data tahap-tahap pengerjaan proyek. Pengumpulan data dilakukan dengan melalui tiga tahapan. Tahap pertama dilakukan proses interview kepada mahasiswa atau alumni Teknik Elektro Universitas Muhammadiyah Malang untuk mendapatkan spesifikasi tahapan pengerjaan. Tahapan kedua yaitu menyusun data yang didapatkan dari hasil interview akan disesuaikan dengan kebutuhan aplikasi untuk dibentuk menjadi data tahapan setiap bidang. Tahapan terakhir yaitu melakukan analisis terhadap data yang diperoleh untuk menjadi data yang siap diolah.

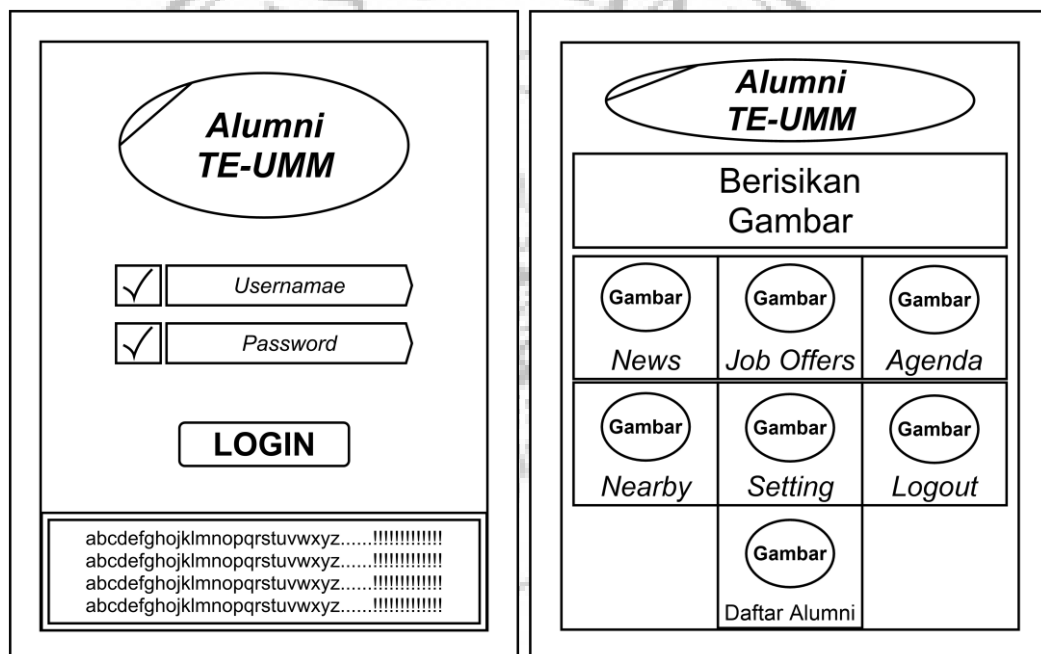
### 3.4 Perancangan Sistem

Pada tahap ini secara garis besar akan dilakukan perancangan untuk mendesain aplikasi alumni Teknik Elektro Universitas Muhammadiyah seperti beberapa gambar sebagai berikut :



Gambar 3.2 Desain Sistem Aplikasi Alumni

Pada gambar 3.2 telah di tunjukan desain penelitian ini yang menunjang berhasilnya pembuatan sistem aplikasi. Aplikasi alumni berbasis android akan di bangun oleh peneliti menggunakan *software android studio* dengan bahasa pemrograman *java*, Sedangkan *website* akan dibangun menggunakan *software CodeIgniter* dengan menggunakan bahasa pemrograman *PHP*. Dari gambar diatas terdapat data base, yaitu merupakan tempat untuk menyimpan data yang sebelumnya sudah dibuat oleh peneliti, yang berfungsi untuk mensuplai data kesistem aplikasi dan website. *Application Programming Interface (API)* merupan cara yang digunakan untuk menghubungkan antara aplikasi dan *web* supaya saling berinteraksi.

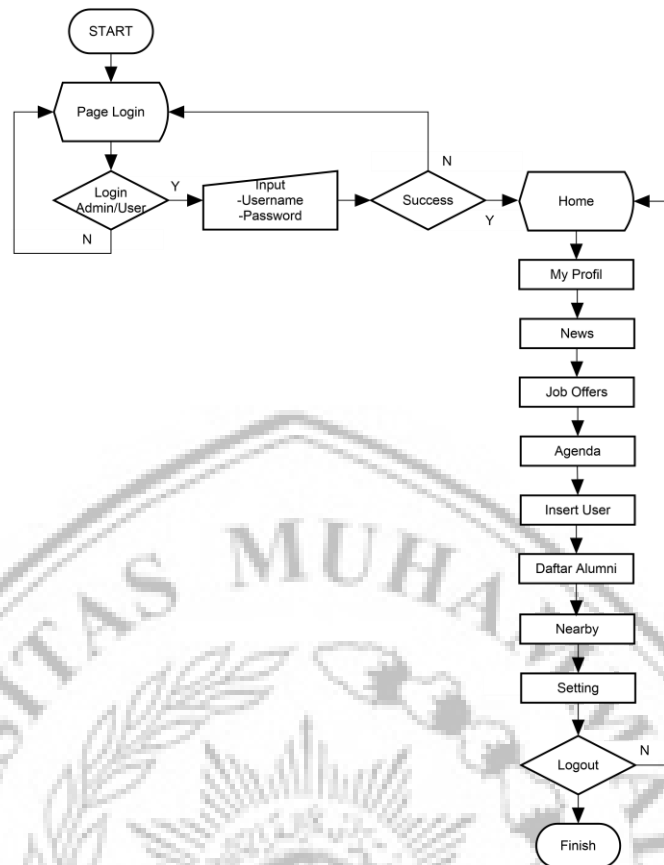


Gambar 3.3 Desain Perancangan Aplikasi Alumni

Pada gambar 3.3 telah di tunjukan desain aplikasi alumni, sebagaimana nantinya peneliti akan merancang dan membangun aplikasi seperti gambar diatas. Selanjutnya akan dijelaskan pengertian dan fungsi masing-masing fitur yang ada pada aplikasi alumni sebagai berikut :

- a) *Home* : Adalah sebagai peran utama untuk menampilkan fitur-fitur menu pada aplikasi.

- b) *My Profil* : Adalah sebuah fitur yang menyimpan data diri sebagai pengguna aplikasi.
- c) *News* : Adalah fitur tempat penyimpanan yang dipakai oleh pengguna aplikasi untuk berbagai pesan yang dikirimkan dari banyak pengguna pada lokasi-lokasi yang berlainan.
- d) *Job Offers* : Adalah fitur tempat berbagi informasi tawaran pekerjaan bagi pengguna aplikasi.
- e) *Agenda* : Adalah fitur untuk mengetahui semua jadwal acara yang akan diselenggarakan oleh pengguna aplikasi.
- f) *Nearby* : Adalah sebuah cara bagi pengguna aplikasi alumni untuk mencari/menemukan pengguna aplikasi lain dalam kondisi berada di lokasi terdekat.
- g) *Insert User* : Adalah tempat untuk memasukan data pengguna baru untuk menjalankan aplikasi maupun *website*. Memasukan data hanya bisa dilakukan oleh admin di *website*.
- h) *Daftar Alumni* : Adalah tempat untuk melihat daftar wisudawan setiap tahunnya, sekaligus pengguna aplikasi dan *website*.
- i) *Settings* : Adalah fitur yang digunakan untuk mengedit akun pada pengguna aplikasi.
- j) *Logout* : Adalah fitur yang digunakan untuk keluar dari aplikasi.



Gambar 3.4 *Flowchart* Sistem Aplikasi

Pada gambar 3.4 *flowchart* sistem aplikasi diatas ditunjukkan bagaimana gambaran tentang sistem aplikasi, diperjelas sebagai berikut :

- User/admin* melakukan *login* di halaman *login* aplikasi.
- User/admin* memasukan *username* dan *password*.
- Jika *username* dan *password* salah, maka sistem secara otomatis kembali ke halaman *login* aplikasi.
- Jika *username* dan *password* benar, maka sistem secara otomatis akan melanjutkan ke halaman utama aplikasi.
- Dari halaman utama aplikasi, jika *user/admin* memilih fitur *my profil*, maka sistem akan menampilkan isi profil pengguna.
- Jika *user/admin* memilih fitur *news*, maka sistem akan menampilkan berbagai pesan yang telah di unggah pengguna aplikasi dan *website* sebelumnya.

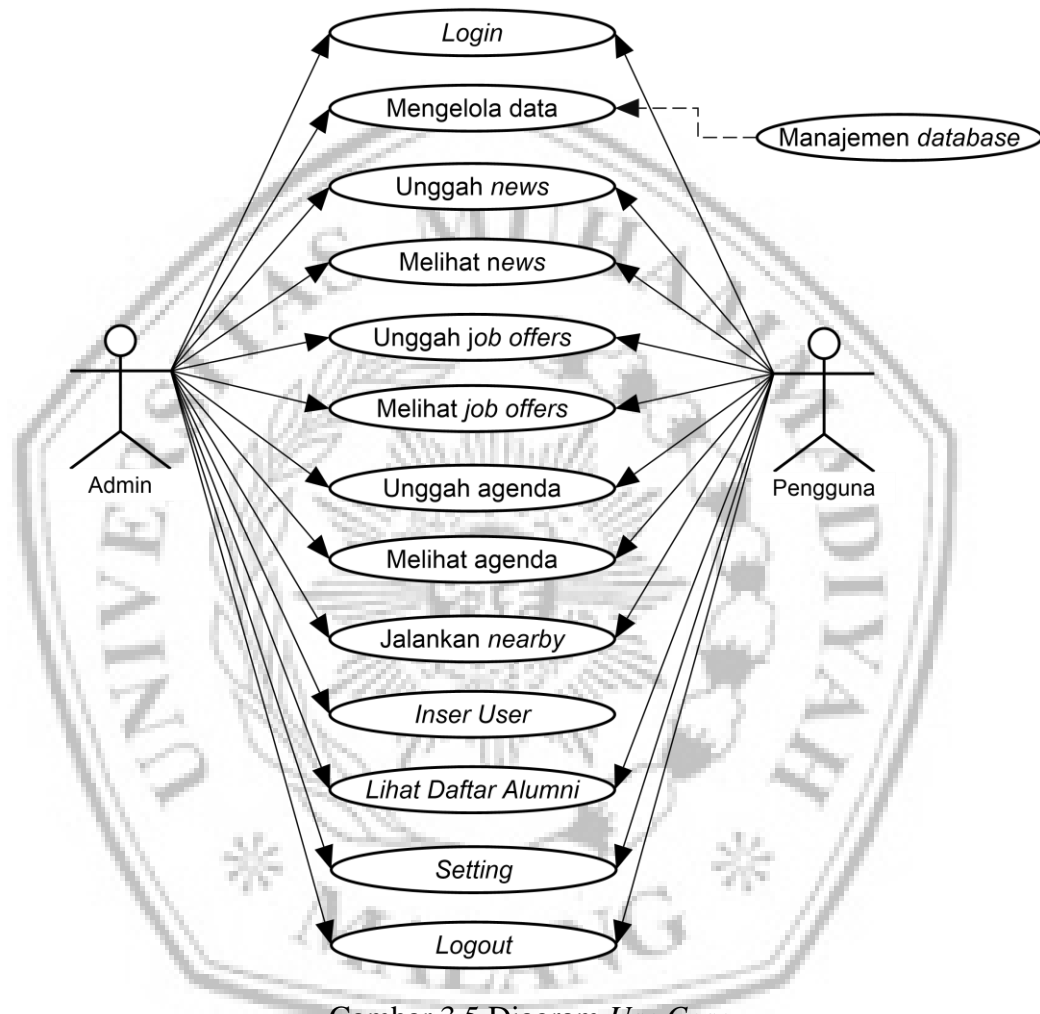
- g) Jika *user/admin* memilih fitur *job offers*, maka sistem akan menampilkan lowongan pekerjaan yang telah di unggah pengguna aplikasi dan *website* sebelumnya.
- h) Jika *user* memilih fitur agenda, maka sistem akan menampilkan halaman jadwal agenda yang telah diunggah pengguna aplikasi dan *website* sebelumnya.
- i) Jika *user/admin* memilih fitur *nearby* maka sistem akan menampilkan/memperlihatkan pengguna yang sedang *online* di area sekitar.
- j) Dari halaman utama admin pada *website*, jika *admin* memilih fitur *insert user*, maka sistem akan menampilkan halaman untuk melakukan tambah akun.
- k) Dari halaman utama aplikasi, jika *user/admin* memilih fitur daftar alumni, maka sistem akan menampilkan isi daftar alumni.
- l) Jika *user/admin* memilih fitur *settings*, maka sistem akan menampilkan halaman *settings* yang dapat digunakan untuk mengedit profil.
- m) Jika *user/admin* memilih fitur *logout*, maka sistem secara otomatis akan menampilkan di halaman *login*.

### **3.5 Perancangan Model Sistem Aplikasi Menggunakan Metodel *UML* (*Unified Modeling Language*)**

*UML (Unified Modeling Language)* adalah metode pemodelan secara *visual* sebagai sarana untuk merancang dan atau membuat *software* berorientasi objek, karena *UML* ini merupakan bahasa visual untuk pemodelan bahasa berorientasi objek, maka semua *elemen* dan diagram berbasiskan pada paradigma *object oriented*. *UML* juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang *spesifik*, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software*. *UML* terdiri dari beberapa diagram, yaitu *usecase diagram*, *class diagram*, *state diagram*, *activity diagram* *sequence diagram*, *collaboration diagram*, *component diagram*, dan *deployment diagram* (Alfarisyi & Amila, 2014).

### 3.5.1 Use Case Diagram

*Use Case Diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut, sehingga pembuatan *use case* diagram lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case* diagram merepresentasikan sebuah interaksi antara aktor dengan sistem (Kalza, 2012).



Gambar 3.5 Diagram Use Case

Skenario dari usecase diatas sebagai berikut :

a) *Use Case* : Mengelolah *Database*.

Aktor : *Admin*.

Pre-Kondisi : *Admin* belum menambahkan data pengguna baru.

Post-Kondisi : *Admin* melakukan tambah data pengguna baru.

Deskripsi : *Use case* ini merupakan proses dimana admin melakukan proses tambah data pengguna aplikasi di dalam *database server*.

Tabel 3.1 Skenario Pengelolaan data di *database*

<i>Admin</i>		<i>Sistem</i>	
1	Membuka halaman <i>database</i>	2	Sistem menampilkan halaman <i>database</i>
3	Menambahkan data pengguna baru		
4	Simpan	5	Sistem menyimpan data pengguna baru

- b) *Use Case* : *Login*.
- Aktor : *Admin/Pengguna*.
- Pre-Kondisi : *Admin/Pengguna* belum memasukkan *username* dan *password*.
- Post-Kondisi : *Admin/Pengguna* memasukkan *username* dan *password*.
- Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses *login* ke dalam aplikasi/*website*.

Tabel 3.2 Skenario Proses *Login*

<i>Admin/Pengguna</i>		<i>Sistem</i>	
1	Membuka halaman <i>login</i>	2	Sistem menampilkan halaman <i>login</i>
3	Memasukan <i>username</i> dan <i>password</i>	4	Sistem menampilkan halaman utama aplikasi



c) *Use Case* : Melihat atau Mengunggah *news*.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum memilih fitur *news*.

Post-Kondisi : *Admin/Pengguna* sudah memilih fitur *news*.

Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses penambahan berita di fitur *news* atau hanya melihat.

Tabel 3.3 Skenario Proses Melihat atau Pengunggahan Berita Baru di Fitur *News*

<i>Admin/Pengguna</i>		Sistem	
1	Membuka halaman <i>news</i>	2	Sistem menampilkan halaman <i>news</i>
3	Melihat atau Mengunggah berita baru di dalam fitur <i>news</i>		
4	Simpan	5	Sistem menyimpan berita baru

d) *Use Case* : Melihat atau Mengunggah *Job Offers*.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum memilih fitur *job offers*.

Post-Kondisi : *Admin/Pengguna* sudah memilih fitur *job offers*.

Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses penambahan lowongan pekerjaan di fitur *job offers* atau hanya melihat.

Tabel 3.4 Sekenario Proses Melihat atau Pengunggahan Lowongan Kerja di Fitur *Job Offers*

<i>Admin/Pengguna</i>		<b>Sistem</b>	
1	Membuka halaman <i>job offers</i>	2	Sistem menampilkan halaman <i>job offers</i>
3	Melihat atau Mengunggah lowongan pekerjaan di dalam fitur <i>job offers</i>		
4	Simpan	5	Sistem menyimpan masukan data lowongan pekerjaan dari pengguna

e) *Use Case* : Melihat atau Mengunggah Agenda.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum memilih fitur agenda.

Post-Kondisi : *Admin/Pengguna* sudah memilih fitur agenda.

Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses penambahan agenda baru di fitur agenda atau hanya melihat.

Tabel 3.5 Sekenario Proses Melihat atau Pengunggahan Agenda di Fitur Agenda

<i>Admin/Pengguna</i>		<b>Sistem</b>	
1	Membuka halaman agenda	2	Sistem menampilkan halaman agenda
3	Melihat atau Mengunggah lowongan pekerjaan di dalam fitur agenda		
4	Simpan	5	Sistem menyimpan masukan agenda baru dari pengguna

f) *Use Case* : Menjalankan *nearby*.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum mengaktifkan *nearby*.

Post-Kondisi : *Admin/Pengguna* sudah mengaktifkan *nearby*.

Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses penggunaan fitur *nearby*.

Tabel 3.6 Skenario Proses Menjalankan Fitur *Nearby*

<i>Admin/Pengguna</i>		<i>Sistem</i>	
1	Membuka fitur <i>nearby</i>	2	Sistem menampilkan halaman <i>nearby</i>
3	Memilih tombol aktif di halaman <i>nearby</i>	4	Sistem mengaktifkan <i>nearby</i>
5	Melihat	6	Sistem menampilkan daftar pengguna lain yang sedang aktif di lokasi terdekat

g) *Use Case* : *Insert User*.

Aktor : *Admin*.

Pre-Kondisi : *Admin* belum memilih fitur *insert user*.

Post-Kondisi : *Admin* sudah memilih fitur *insert user*.

Deskripsi : *Use case* ini merupakan proses dimana *admin* melakukan proses penambahan akun baru di fitur *insert user*.

Tabel 3.7 Sekenario Proses Penambahan Akun Baru Di Fitur *Insert User*

<i>Admin/Pengguna</i>		<b>Sistem</b>	
1	Membuka halaman <i>insert user</i>	2	Sistem menampilkan halaman <i>insert user</i>
3	Melihat atau Mengunggah akun baru di dalam fitur <i>insert user</i>		
4	Simpan	5	Sistem menyimpan masukan akun baru dari <i>admin</i>

h) *Use Case* : Melihat Daftar Alumni Pada Aplikasi.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum memilih fitur daftar alumni.

Post-Kondisi : *Admin/Pengguna* sudah memilih fitur daftar alumni.

Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melihat daftar alumni.

Tabel 3.8 Sekenario Proses Melihat Fitur Daftar Alumni

<i>Admin/Pengguna</i>		<b>Sistem</b>	
1	Membuka fitur daftar alumni	2	Sistem menampilkan halaman daftar alumni.

i) *Use Case* : Melakukan *settings*.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum memilih fitur *setting*.

Post-Kondisi : *Admin/Pengguna* sudah memilih fitur *setting*.

Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses *setting* pada *akun*.

Tabel 3.9 Sekenario Proses *Settings*

<i>Admin/Pengguna</i>		<i>Sistem</i>	
1	Membuka fitur <i>settings</i>	2	Sistem menampilkan halaman setting, tetapi berisikan halaman <i>my profil</i>
3	Mensetting akun		
5	Simpan	6	Sistem menyimpan hasil setingan dari pengguna

j) *Use Case* : *Logout*.

Aktor : *Admin/Pengguna*.

Pre-Kondisi : *Admin/Pengguna* belum memilih fitur *logout*.

Post-Kondisi : *Admin/Pengguna* sudah memilih fitur *logout*.

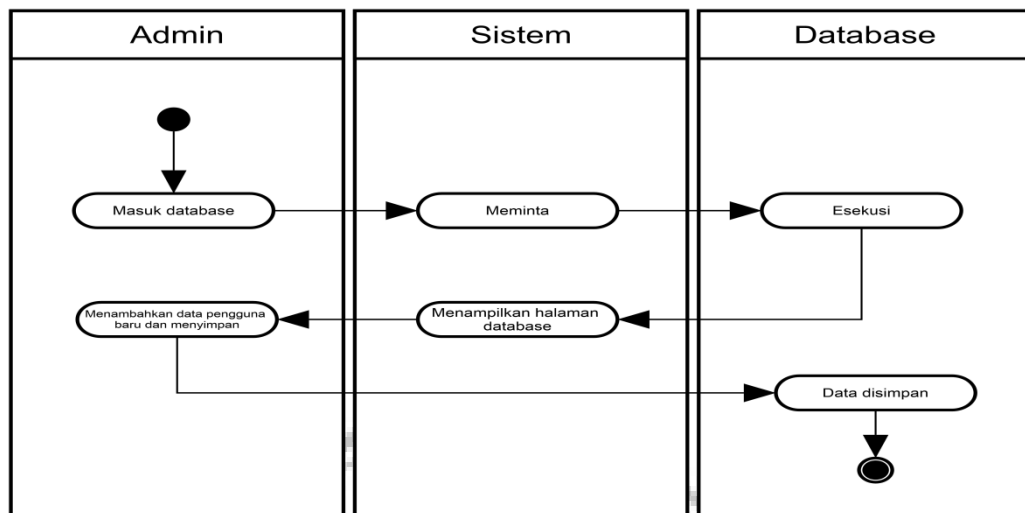
Deskripsi : *Use case* ini merupakan proses dimana *admin/pengguna* melakukan proses *logout* dari aplikasi.

Tabel 3.10 Sekenario Proses *Logout*

<i>Admin/Pengguna</i>		<i>Sistem</i>	
1	Memilih fitur <i>logout</i>	2	Sistem otomatis keluar dan menampilkan halaman utama.

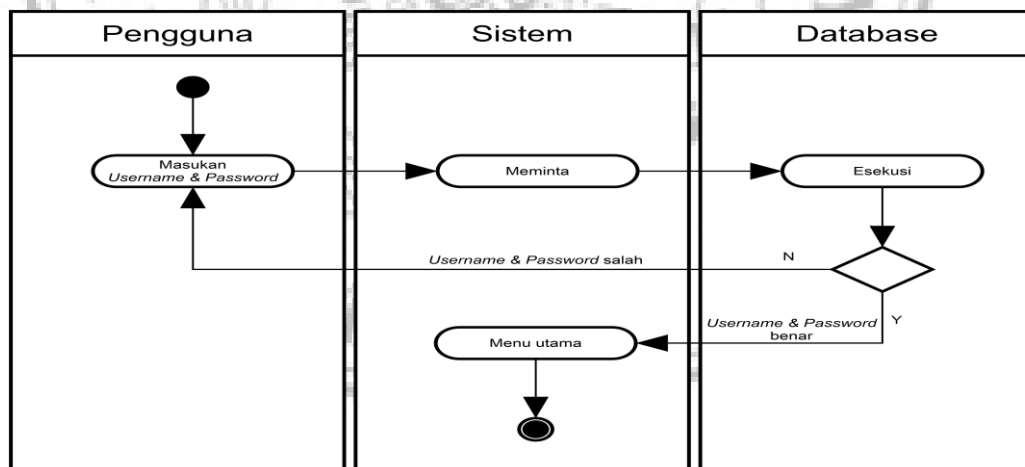
### 3.5.2 *Activity Diagram*

*Activity Diagram* merupakan gambaran detail dan *use case diagram* dimana setiap state merupakan suatu aksi (*action state*) dan transisinya dipicu oleh aksi (*action*) yang sudah selesai dan state sebelumnya dan biasanya digunakan untuk menunjukkan urutan dan *state-state*. Berikut adalah *Activity Diagram* dari pendaftaran ke aplikasi.



Gambar 3.6 Diagram Activity Pengelolaan Database

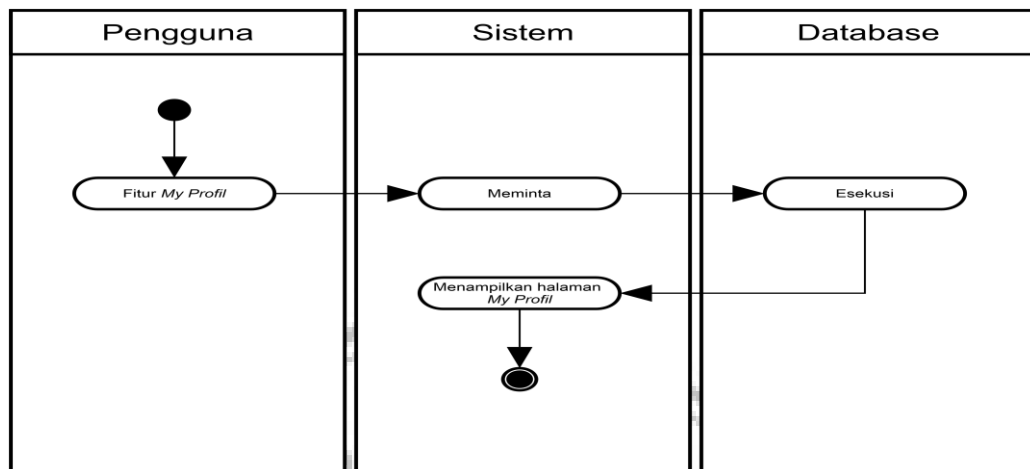
Pada Gambar 3.6 dijelaskan tentang *activity diagram* pengelolaan database yang dilakukan oleh admin. Pertama yang dilakukan admin adalah masuk ke database terlebih dahulu dengan cara [http://localhost/phpmyadmin/\(database\)](http://localhost/phpmyadmin/(database)), secara otomatis sistem akan menampilkan halaman database, setelah itu admin menambahkan dan menyimpan data, maka data akan tersimpan di database.



Gambar 3.7 Diagram Activity Login

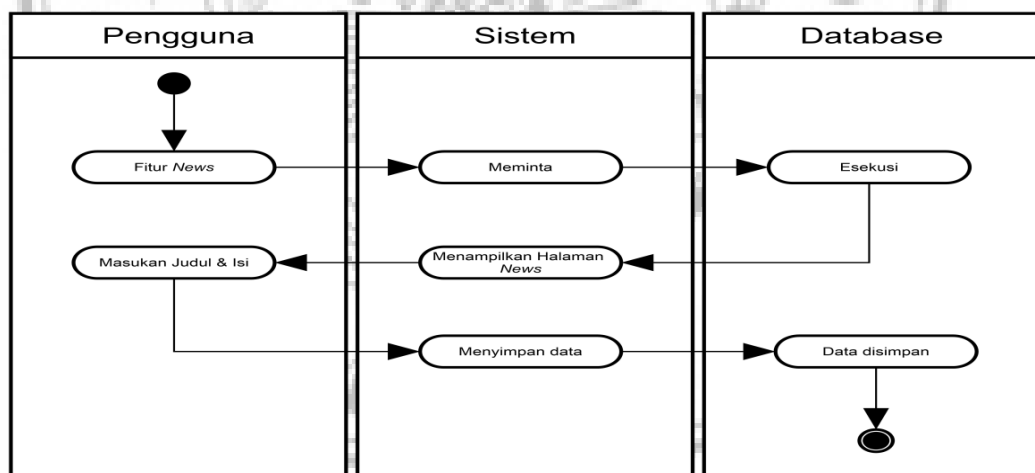
Pada Gambar 3.7 dijelaskan tentang *activity diagram login* dari aplikasi yang dijalankan oleh *user*, disini *user* akan diminta oleh sistem untuk memasukan *username* dan *password* yang sudah disimpan di database, jika data yang

dimasukan benar maka sistem akan menampilkan halaman utama aplikasi dan jika data tidak benar maka sistem akan mengembalikanya ke halaman awal *login*.



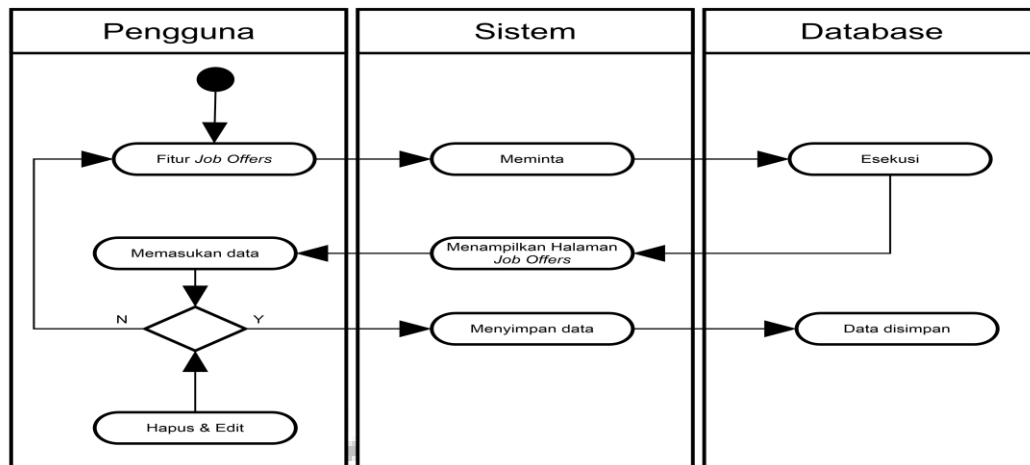
Gambar 3.8 Diagram Activity My Profil

Pada Gambar 3.8 dijelaskan tentang *activity diagram fitur my profil*, dimana sistem meminta data ke database yang nantinya akan menampilkan halaman *fitur my profil* dan berisikan profil lengkap pengguna aplikasi.



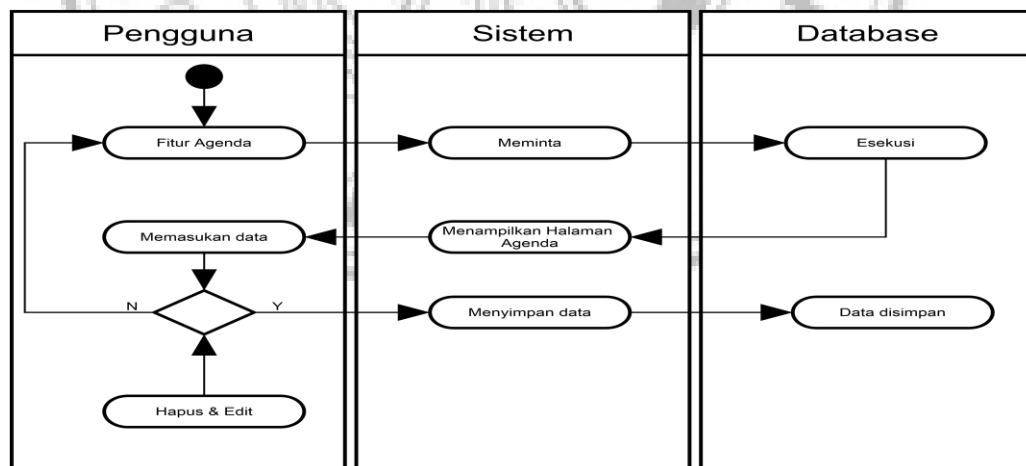
Gambar 3.9 Diagram Activity News

Pada Gambar 3.9 dijelaskan tentang *activity diagram news*, dimana sistem meminta data ke database, dan dieksekusi oleh database lalu dikembalikan ke sistem untuk menampilkan halaman *fitur news*. Pada tampilan halaman *fitur news* pengguna mengisi judul dan isi, setelah diklik kirim maka data akan disimpan di database.



Gambar 3.10 *Diagram Activity Job Offers*

Pada Gambar 3.10 dijelaskan tentang *activity diagram job offers*, dimana sistem meminta data ke database, dan dieksekusi oleh database lalu dikembalikan ke sistem untuk menampilkan halaman *fitur job offers*. Jika pengguna ingin menambahkan atau mengedit dan menghapus data maka, pengguna harus mengisi atau mengedit dan menghapus data yang sudah disiapkan oleh sistem dan di simpan. Setelah data disimpan maka sistem akan mengirim ke database untuk disimpan. Jika tidak maka sistem hanya menampilkan halaman *fitur job offers*.

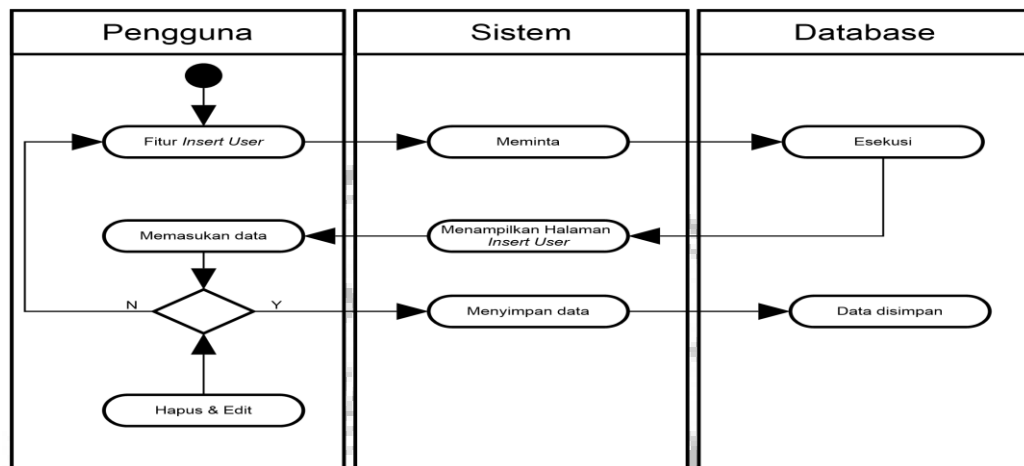


Gambar 3.11 *Activity Diagram Agenda*

Pada Gambar 3.11 dijelaskan tentang *activity diagram agenda*, dimana sistem meminta data ke database, dan dieksekusi oleh database lalu dikembalikan

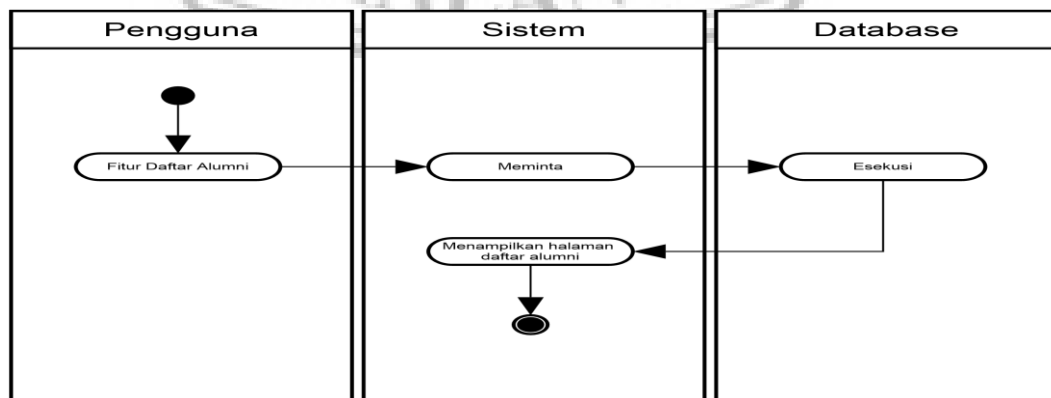


ke sistem untuk menampilkan halaman fitur agenda. Jika pengguna ingin menambahkan atau mengedit dan menghapus data maka, pengguna harus mengisi atau mengedit dan menghapus data yang sudah disiapkan oleh sistem dan di simpan. Setelah data disimpan maka sistem akan mengirim ke database untuk disimpan. Jika tidak maka sistem hanya menampilkan halaman fitur agenda.



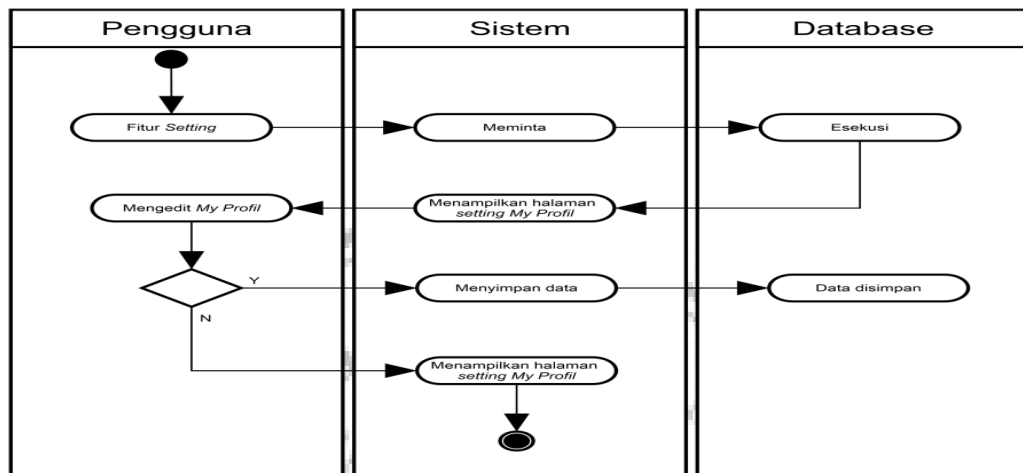
Gambar 3.12 Activity Diagram Insert User

Pada Gambar 3.12 dijelaskan tentang *activity diagram insert user*, dimana sistem meminta data ke database, dan dieksekusi oleh database lalu dikembalikan ke sistem untuk menampilkan halaman fitur *insert user*. Jika admin ingin menambahkan atau mengedit dan menghapus data maka, admin harus mengisi atau mengedit dan menghapus data yang sudah disiapkan oleh sistem dan di simpan. Setelah data disimpan maka sistem akan mengirim ke database untuk disimpan. Jika tidak maka sistem hanya menampilkan halaman fitur *insert user*.



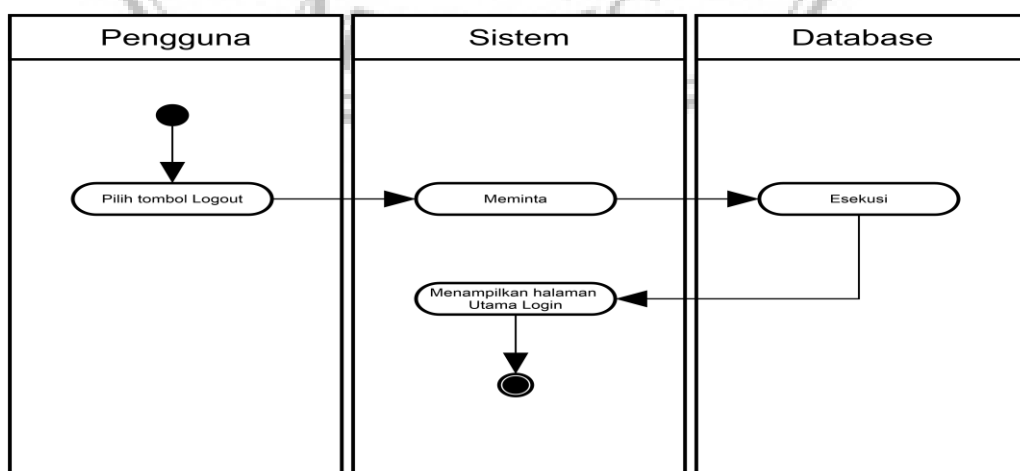
Gambar 3.13 Diagram Activity Daftar Alumni

Pada Gambar 3.13 dijelaskan tentang *activity diagram fitur* daftar alumni, dimana sistem meminta data ke database yang nantinya akan menampilkan halaman *fitur* daftar alumni dan berisikan daftar alumni.



Gambar 3.14 Activity Diagram Settings

Pada Gambar 3.14 dijelaskan tentang *activity diagram settings*, dimana sistem meminta data ke database, dan dieksekusi oleh database lalu dikembalikan ke sistem untuk menampilkan halaman *fitur setting* tetapi berisikan data *my profil*. Jika pengguna ingin mengedit *my profil*, maka setelah data diedit pengguna harus menyimpannya. Setelah data disimpan maka sistem akan mengirim ke database untuk disimpan. Jika tidak maka sistem hanya menampilkan halaman *fitur settings*.

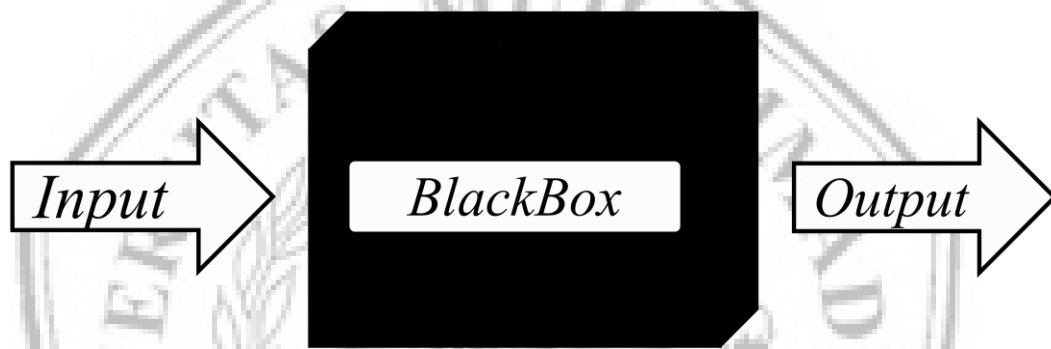


Gambar 3.15 Activity Diagram Logout

Pada Gambar 3.15 dijelaskan tentang *activity diagram logout*, dimana jika pengguna ingin mengakhiri proses pemakaian aplikasi, maka pengguna memilih tombol *logout* dan sistem akan meminta ke database, dan di esekusi database untuk dikembalikan ke sistem untuk menampilkan halaman utama aplikasi.

### 3.6 Pengujian Sistem

Pengujian aplikasi ini dilakukan dengan cara menguji fungsi-fungsi yang terdapat di dalam sistem.



Gambar 3.16 *BlackBox Testing*

Pengujian ini menggunakan metode *BlackBox testing* yang bertujuan untuk mengetahui keakuratan informasi yang terdapat pada aplikasi terhadap kebutuhan dari *entitas-entitas* yang terkait. Setelah itu, dilakukan analisis terhadap perubahan-perubahan yang terjadi pada aplikasi alumni Teknik Elektro Universitas Muhammadiyah Malang dengan diterapkan sistem berbasis android dan web. Analisis yang dilakukan mencakup analisis antara proses layanan sistem yang terdapat di aplikasi dan analisis perpindahan informasi.